**Bifid Decrypt**

**Background**

Encryption algorithms are a set of instructions to turn a plaintext (i.e. ordinary message) into an unreadable message (i.e. ciphertext). Decryption algorithms are a set of instructions that turn the ciphertext back into the original plain text (sometimes it is a step-by-step reversal of an encryption algorithm but not always). The bifid cipher uses a matrix of characters where the rows and columns of each character are used to encrypt the message. The matrix is called a polybius square in the context of cryptography. You will be tasked with implementing the decryption algorithm for the bifid cipher.

**Problem**

In a math class, your group is given a 6x6 matrix where it contains 26 uppercase characters and 10 numbers from 0-9 where each is encoded as a string. You are asked to use the polybius square to decrypt a message which you are given by the instructor. The teacher gives you a general rule to encrypt messages so you’ll have to figure out how to reverse it. You know that for every message, the position of the character will be written down in an order that the numbers of rows of all the characters and then the numbers of columns of all characters. For example,the polybius square you’re given is

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Column** | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** |
| **Row** | **0** | “B” | “G” | “0” | “W” | “K” | “Z” |
| **1** | “Q” | “1” | “P” | “L” | “D” | “S” |
| **2** | “9” | “I” | “O” | “X” | “7” | “E” |
| **3** | “F” | “4” | “C” | “U” | “M” | “6” |
| **4** | “T” | “H” | “2” | “V” | “R” | “5” |
| **5** | “N” | “A” | “8” | “Y” | “3” | “J” |

and the plaintext is “HELLO”. The order of numbers that is written down is

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **H** | **E** | **L** | **L** | **O** |
| **Row** | 4 | 2 | 1 | 1 | 2 |
| **Column** | 1 | 5 | 3 | 3 | 2 |

The row and then columns will be added together to form [4,2,1,1,2,1,5,3,3,2]. Then, the numbers will be put into groups of 2 from left to right to get a new message. In this case, the message will be shown as (4,2),(1,1),(2,1),(5,3),(3,2) which will give you the ciphertext of “21IYC”. You now need to figure out how to go from “21IYC” to “HELLO”.

**Constraints**

Note: We won’t use any character that is outside the matrix, but any characters not in the matrix(e.g. !, ?, & ) must be preserved. The length of the message will range from 0 - 100. Only uppercase characters like in the matrix will be used. Every character in the polybius square is a string.

|  |  |
| --- | --- |
| **Input (list polybiusSquare, string ciphertext)** | **Output(string plaintext)** |
| 21XJY, OS 7AOK 3I 8MJC SU7! FDAJ IK E4QS CMG5? | HELLO, MY NAME IS JOHN DOE! WHAT IS YOUR NAME? |
| RIA Q7D72542 QD YVE 9SIJC R0 : GYLFSI7S39 | THE PASSWORD TO THE VAULT IS : 0123456789 |
| 77NIA7I2E V2FEY 3F 7UX J0A017N! | EVERYBODY LOVES TO EAT CHICKEN! |
| R7J 1J79 7ZSM61YP3 GCG AQ5QQS34 A94UC 6DL6. | THE HAND SANITIZER WAS ACTUALLY CLEAR GLUE. |

**Testing**

A Python unit test can be found alongside the bifidDecrypt.py file. After filling in the method definition inside bifidDecrypt.py, run the test. To run the test, either run the unit tests in an IDE or in the command line, cd into the directory with the file and type <test\_bifidDecrypt.py>.